90-06638/

CRT-694

SCHEDULING TASKS AND VEHICLES IN A FLEXIBLE MANUFACTURING SYSTEM

by

Jacek Blacewicz*
Horst A. Eiselt**
Gerd Finke***
Gilbert Laporte****
Jan Weglarz*

* Instytut Informatyki, Automatyki i Robotyki, Politechnika Poznańska, Poznań, Poland.

** Faculty of Administration, University of New Brunswick, Fredericton, Canada.

*** Laboratoire Artemis, Université de Grenoble, Grenoble, France.

**** Centre de recherche sur les transports, Université de Montréal, Montréal, Canada.

Centre de recherche sur les transports - Publication #694

May 1990

Dépôt légal - 2e trimestre 1990

ABSTRACT

Due to their increasing applicability in modern industry, flexible manufacturing systems (FMS), their design and control have been studied extensively in the recent literature. One of the most important issues arising in this context is the FMS scheduling problem. This paper is concerned with a new model of an FMS system, motivated by the practical application that takes into account both machine and vehicle scheduling. For the case of a given machine schedule a simple polynomial time algorithm is presented that checks the feasibility of a vehicle schedule and constructs it whenever one exists. Then a dynamic programming approach is proposed constructing optimal machine and vehicle schedules. This technique results in a pseudopolynomial time algorithm for a fixed number of machines.

Key words: flexible manufacturing systems; production scheduling; vehicle scheduling.

- - - - -

RÉSUMÉ

En raison de l'applicabilité croissante des systèmes de production flexibles dans l'industrie, on a assisté récemment à une éclosion d'articles sur leur conception et sur le contrôle. Les problèmes d'ordonnancement de production issus de tels contextes sont particulièrement importants. On étudie dans cet article un problème motivé par une situation réelle, où l'on doit à la fois déterminer l'horaire de production et celui des véhicules chargés d'acheminer les pièces aux machines. On décrit un algorithme polynomial qui, pour un horaire de production donné, construit un horaire de véhicules réalisable, lorsqu'il en existe un. On propose par la suite un algorithme de programmation dynamique pour la construction simultanée de l'horaire de production et de celui des véhicules. Il en résulte un algorithme pseudopolynomial pour un nombre fixe de machines.

Mots-clés : systèmes de production flexibles; horaires de production; horaires de véhicules.

1. Introduction

The increasing role of flexible manufacturing systems (FMS) in modern industry creates the need to analyse their behavior and to work out methods of their design and control. There are several questions arising in this context. According to [17,19] they may be divided into several groups, most importantly: design problems, planning problems, scheduling problems as well as control and monitoring problems. The present paper concerns the scheduling problem and is motivated by the practical problems associated with the implementation of a flexible manufacturing system by a major North-American manufacturer of helicopter parts.

The large number of papers in journals and special volumes on FMS devoted to the scheduling problem (see eg. [13,14,16,17,21,22] reflects the importance of that area for the efficient utilization of a FMS system. However, as pointed out in [12], the majority of these papers dealt with either part and machine scheduling [1,4,5,7,8,18,20] or with Automated Guided Vehicle (AGV) routing, eg. [2,3,24] separately. One of the few exceptions is [15] where besides machines also tools, fixtures and pallets have been taken into account.

In the present paper both issues, i.e. part and machine scheduling and vehicle routing are considered together within a framework of FMS scheduling. The objective is to construct a schedule of minimum length, which includes both issues. The organization of the paper is as follows. Sections 2 contains the description and modelling of a certain flexible manufacturing system. In section 3 a procedure for vehicle routing is presented, assuming that a production schedule is given. Section 4 deals with simultaneous task and vehicle scheduling based on a dynamic programming procedure. Some conclusions and possible extensions are pointed out in section 5.

2. Description and modelling of the system

The flexible manufacturing system under consideration has been implemented by one of the manufacturers producing parts for helicopters. A schematic view of the system is presented in Fig.1 and its description is as follows.

Pieces of raw material from which the parts are machined are stored in the automated storage area AS (1). Whenever necessary, an appropriate piece of material is taken from the storage and loaded onto the pallet and vehicle at the stand (2). This operation is performed automatically by computer controlled robots. Then, the piece is transported by an automated guided vehicle AGV (7) to the desired machine (6) where it is automatically unloaded at (8). Every machine in the system is capable of processing any machining operation. This versatility is achieved by a large number of tools and fixtures that may be used by the machines. The tool-magazines (4) of every machine have a capacity of up to 130 tools which are used for the various machining operations. The tools of the magazines are arranged in two layers so that the longer tools can occupy two vertical positions. The tools and, to a lesser extent, fixtures are changed automatically. It should be noted that a large variety of almost 100 quite different parts can be produced by each of these machines in this particular FMS. Simpler parts require about 30 operations (and tools) and the most complicated parts need in the order of 80 operations. Therefore the tool-magazines have sufficient capacity to stock the tools for one to several consecutive parts in a production schedule. In addition the tools are loaded from a large automated central tool storage area (3) which is located closely to the machines. No tool competition is observed, since the storage area contains more than 2000 tools (including many multiple tools) and there are just 4 NC-machines. The delivered raw material is mounted onto the appropriate fixture and processed by the tools which are changed according to a desired plan. The tool technology of this particular system allows the changing of the tools during execution of the tasks.

This is used to eliminate the set-up times of the tools required for the next task and occasionally a transfer of a tool to another machine (to validate completely the no-resource competition). The only (negligible) transition time in thie FMS that could be observed was in fact the adjustment in size of the spindle that holds the tool whenever the next tool is exchanged with the previous one. After the completion the finished part exchanges its position with the raw material of the next task that is waiting for its processing. It is then automatically transported by an AGV to the inspection section (9). Parts which passed the inspection are transported and unloaded at the storage area (10).

We see that the above system is very versatile and this feature is gained by the usage of many tools and large tool magazines. As pointed out in [11,12] it is a common tendency of modern flexible manufacturing systems to become so versatile that most of the processes on a part can be accomplished by just one or at most two machine types. Thus, as the result one gets a system consisting of identical parallel machines. On the other hand, the existence of a large number of tools in the system allows one not to consider resource (tool) competition. Hence our problem reduces in fact to that of simultaneous scheduling and routing of parts among parallel machines. The inspection stage can be postponed in that analysis, since it is performed separately on the first-come-first-served basis.

Following the above observations, we can model the considered flexible manufacturing system using elements described below. Given a set of n *independent tasks* (parts) $T_1, T_2, \ldots, T_n$ that are to be processed on a set of m parallel identical *machines* $P_1, P_2, \ldots, P_m$ (m being not a very large number). Here parallelism means that every machine is capable of processing any task. For its processing, every task T_j requires one arbitrary machine and specified amounts of *additional resources* [2]. Here we distinguish between *renewable resources* referring to different tools used throughout a processing of the task and *nonrenewable* resources which represent raw material. Task T_j should be granted one machine and the required resources

during the noninterrupted time interval of length p_j. The last value is called a *processing time* of task T_j. From the preceding remarks one may assume that there are enough resources of both categories and the only scarce resource the tasks should compete for are the machines. Set-up times connected with changing tools are assumed to be zero since the latter can be changed on-line during the execution of tasks. Set-up times resulting from changing part fixtures are included in the processing times.

As mentioned above, machines are identical except for their locations and thus they require different delivery times. Hence, we may assume that k ($k \leq m$) *automated guided vehicles* (AGV) $V_1, V_2, \ldots, V_k$, are to deliver pieces of raw material from the storage area to specified machines and the *time* associated with the *delivery* is equal to τ_i, $i=1,2,\ldots,m$. The delivery time includes *loading time* at the storage area and *unloading time* at the required machine, both times being equal to *a*. During each trip exactly one piece of raw material is delivered; this is due to the dimension of parts to be machined. After delivery of a piece of raw material the vehicle takes a pallet with a processed part (maybe from another machine), delivers it to the inspection stage and returns to the storage area (1). The round trip takes A units of time, including two loading and two unloading times. It is apparent that the most efficient usage of vehicles in the sense of a throughput rate for pieces delivered is achieved when the vehicles are operating at a cyclic mode with cycle time equal to A. In order to avoid traffic congestions we assume that starting moments of consecutive vehicles at the storage area are delayed by *a* time units.

The problem is now to construct a *schedule* for machines and vehicles such that the whole task set is processed in a minimum time.

It is obvious that the general problem stated above is NP-hard, as it is already NP-hard for the nonpreemptive scheduling of two machines (cf [13]). In the following two sections we will consider two variants of the problem. In the first, the production schedule (i.e. the assignment of tasks to machines) is assumed to be known and the objective is to find a

feasible schedule for vehicles. This problem can be solved in polynomial time. The second problem consists of finding a composite schedule, i.e. one taking into account simultaneous assignment of vehicles and machines to tasks.

3. Vehicle scheduling for a fixed production schedule

In this section we consider the problem of vehicle scheduling given a production schedule. Suppose an (optimal) nonpreemptive assignment of tasks to machines is given (cf. Fig.2). This assignment imposes certain deadlines d_j^i on delivery of pieces of raw material to particular machines, where d_j^i denotes the latest moment by which raw material for part T_j should be delivered to machine P_i. The lateness in delivery could result in exceeding the planned schedule length C. Below we describe an approach that allows us to check whether it is possible to deliver all the required pieces of raw material to their destinations (given some production schedule), and if so, a vehicle schedule will be constructed. Without loss of generality we may assume that at time 0 at every machine there is already a piece of material to produce the first part; otherwise one should appropriately delay starting times on consecutive machines.

Our vehicle scheduling problem may be now formulated as follows. Given a set of deadlines d_j^i, $j=1,2,\ldots,n$ and delivery times from the storage area to particular machines τ_i, $i=1,2,\ldots,m$, is that possible to deliver all the required pieces of raw material on time, i.e. before the respective deadlines. In the case of a positive answer a feasible vehicle schedule should be constructed.

It follows from the assumptions in section 2 that our vehicle schedule will be a cyclic one, as shown in Fig.3. We see that starting times for vehicles are fixed, but arrival times depend on the particular machines to which pieces of raw material are delivered. In general, the scheduling problem with the objective of meeting deadlines is NP-hard for arbitrary processing (delivery) times and two or more processing (transportation) means. However, in our problem we can use the

cyclic property of the schedule for relatively easy checking the feasibility condition of the vehicle schedule for a given production schedule. The first schedule does not need to be constructed. This is given in Lemma 1.

Lemma 1

For a given set of latest transportation starting times s_j, $s_j \leq s_{j+1}$ $j=1,2,\ldots,n$, one can construct a feasible transportation schedule for k vehicles if and only if:

$$s_j \geq \left(\lceil \frac{j}{k} \rceil - 1\right)A + \left[j - \left(\lceil \frac{j}{k} \rceil - 1\right)k - 1\right]a \quad [1]$$

for all $j=1,2,\ldots,n$.

Proof

It is not hard to prove the correctness of the above formula taking into account that its two components reflect respectively, time necessary for an integer number of cycles and the delay of an appropriate vehicle in a cycle needed for a transportation of the j'th task in order.

■

Conditions given in Lemma 1 can be checked in O(n logn) time in the worst case. If one wants to construct a feasible schedule the following polynomial time algorithm will find it, whenever one exists. The basic idea behind the algorithm is to choose for a transportation such a piece of raw material which corresponds at a given moment to a part whose deadline decreased by the corresponding delivery time is the minimum. It is thus the most urgent delivery at this moment. This approach is summarized by the following algorithm.

---

[1] $\lceil x \rceil$ denotes the smallest integer not smaller than x

Algorithm 1

1. Set t:=0, l:=0.

For all the tasks (pieces of raw material) to be delivered to machines calculate their *latest transportation starting times*:

$$s_j := d_j^i - \tau_i$$

2. At moment t when a vehicle is ready for loading at the storage area, consider the set of remaining tasks and calculate their *slack times*:

$$sl_j := s_j - t.$$

If all are non-negative go to step 3, otherwise no feasible vehicle schedule exists; stop.

3. Choose a task with the minimum value of sl_j and load it onto the vehicle.

Set l:=l+1. If $l \leq k-1$ then t:=t+a otherwise t:=t-(k-1)a+A and l:=0.

If there are any pieces of raw material non-delivered to corresponding machines, then go to step 2, otherwise stop.

A basic property of Algorithm 1 is proved in the following theorem.

Theorem 1

Algorithm 1 finds a feasible transportation schedule whenever one exists.

Proof

Suppose that Algorithm 1 failed to find a feasible transportation schedule while such a schedule S exists. In this case there must exist in S two pieces of raw material T_i and T_j such that $sl_i < sl_j$ and T_j has been transported first. It is not hard to see that exchanging these two pieces, i.e. T_i being transported first, we do not cause the infeasibility of the schedule. Now we can repeat the above pattern as long as such a pair of pieces of raw

material violating the earliest slack time rule exists. After a finite number of such changes one gets a feasible schedule constructed according to Algorithm 1, which is a contradiction.

■

Let us now calculate the complexity of Algorithm 1. considering the off-line performance of the algorithm, then its most complex function is the ordering of tasks in non-decreasing order of their slack times. Thus, the overall complexity would be O(n logn). However, if one performs the algorithm in the on-line mode, then the selection of a task to be transported next requires only linear time , provided that an unordered sequence is used. In both cases a low order polynomial-time algorithm is obtained. We see that the easiness of the problem depends mainly on its regular structure following the cyclic property of the vehicle schedule.

To illustrate the use of the algorithm consider the following example. Let the number of machines m, the number of tasks n and the number of vehicles be equal to 3, 9 and 2, respectively. Transportation times for respective machines are $\tau_1 = 1$, $\tau_2 = 1.5$, $\tau_3 = 2$ and cycle and loading times are A=3, a=0.5, respectively. A production schedule is given in Fig.4a. Thus the deadlines are $d_5^1 = 3$, $d_7^1 = 7$, $d_6^2 = 6$, $d_8^2 = 7$, $d_4^3 = 2$, $d_9^3 = 8$. They result in the latest transportation starting times $s_4 = 0$, $s_5 = 2$, $s_6 = 4.5$, $s_7 = 6$, $s_8 = 5.5$, $s_9 = 6$. The corresponding vehicle schedule generated by Algorithm 1 is shown in Fig.4b. Task T_9 is delivered too late and no feasible transportation schedule for the given production plan can be constructed.

The obvious question is now what to do if there is no feasible transportation schedule. The first approach consists in finding tasks in the trasportation schedule that can be delayed without lengthening the schedule. If such a task is found other tasks that cannot be delayed are transported first. In our example (Fig.4a) task T_7 can be started later and instead T_8 can be assigned first to vehicle V_1. Such an exchange will not

lengthen the schedule. However, it may be also the case that the production schedule reflects deadlines which cannot be exceeded and the tasks cannot be shifted. In such a situation one may use an alternative production plan, if one exists. As pointed out in [17] it is often the case at the FMS planning stage that several such plans may be constructed and the operator chooses one of them. If none can be realized because of a non-feasible transportation schedule, the operator may decide to construct optimal production and vehicle schedules at the same time. One of such approaches based on dynamic programming is described in the next section.

4. Simultaneous task and vehicle scheduling

In this section the problem of simultaneous construction of production and vehicle schedules is discussed. As mentioned above this problem is NP-hard, although not strongly NP-hard, thus, a pseudopolynomial-time algorithm based on dynamic programming can be constructed for its solution.

Assume that tasks are ordered in non-increasing order of their processing times, i.e. $p_1 \geq p_2 \geq \ldots p_{n-1} \geq p_n$.
Such an ordering implies that longer tasks will be processed first and processing can take place on machines with longer distances from the storage area; a fact that is convenient from the viewpoint of vehicle scheduling.

Now let us formulate a dynamic programming algorithm using the ideas presented in [14].

Define

$$x_j(t_1,t_2,\ldots,t_m) = \begin{cases} \underline{\text{true}}, \text{ if tasks } T_1,T_2,\ldots,T_j \text{ can be scheduled on machines } P_1,P_2,\ldots,P_m \text{ in such a way that } P_i \text{ is busy in time interval } [0,t_i],\ i=1,2,\ldots,m \text{ (excluding possible idle time following from vehicle scheduling), and the vehicle schedule is feasible} \\ \underline{\text{false}}, \text{ otherwise} \end{cases}$$

Where

$$x_o(t_1,t_2,\ldots,t_m) = \begin{cases} \underline{\text{true}}, & \text{if } t_i=0,\ i=1,2,\ldots,m \\ \underline{\text{false}}, & \text{otherwise} \end{cases}$$

Using these variables the recursive equation can be written in the following form

$$x_j(t_1,t_2,\ldots,t_m) = \bigvee_{i=1}^{m} \left[x_{j-1}(t_1,t_2,\ldots,t_{i-1},t_i-p_j,t_{i+1},\ldots,t_m) \wedge \right.$$

$$\left. \wedge\ Z_j(t_1,t_2,\ldots,t_{i-1},t_i-p_j,t_{i+1},\ldots,t_m) \right]$$

where

$$Z_j(t_1,t_2,\ldots,t_i-p_j,t_{i+1},\ldots,t_m) = \begin{cases} \underline{\text{true}}, & \underline{\text{if}}\ t_i-p_j-\tau_i \geq (\lceil \frac{j}{k} \rceil -1)A + \\ & + [j-(\lceil \frac{j}{k} \rceil -1)k-1]\,a \\ & \underline{\text{or}}\ j \leq m \\ \underline{\text{false}}, & \text{otherwise} \end{cases}$$

is the condition of vehicle schedule feasibility, given in Lemma 1.

Values of $x_j(\cdot)$ are computed for $t_i=0,1,\ldots,C$; $i=1,2,\ldots,m$, where C is an upper bound on the minimum schedule length C^*_{max}. Finally, C^*_{max} is determined as

$$C^*_{max} = \min \{ \max \{ t_1,t_2,\ldots,t_m \} : x_n(t_1,t_2,\ldots,t_m) = \underline{\text{true}} \}$$

The above algorithm solves our problem in $O(n^2C^m)$ time. Thus, for fixed m, it is a pseudopolynomial-time algorithm, and can be used in practice, taking into account that m is rather small.

To complete our discussion let us consider once more the example from section 3. The above dynamic programming approach yields schedules presented in Fig.5. We see that it is possible to complete all the tasks in 11 units and deliver them to machines in 8 units.

5. Conclusions

In the paper the model of a flexible manufacturing system has been introduced, taking into account both machine and vehicle scheduling. Minimization of the maximum task completion time was the criterion. A simple polynominal time algorithm can be used to find a feasible vehicle schedule (whenever one exists), provided a production schedule is given. For a fixed number of machines,pseudopolynomial dynamic programming approach solves both problems constructing optimal production and vehicle schedules. A various extensions of the model are possible and worth considering. Among them are those including different routes for particular vehicles, an inspection phase as the second stage machine, resource competition and different criteria (e.g. mean flow time or maximum lateness). These issues are currently under investigation.

References

[1] Afentakis, P., An optimal scheduling strategy for flexible manufacturing systems, Working paper No 85-012. Dept. of Industrial Engineering and Operations Research, Syracuse University, N.Y., 1985.

[2] Błażewicz, J., Cellary, W., Słowiński, R., Węglarz, J., Scheduling under Resource Constraints: Deterministic Models, (Annals of Operations Research 7), J.C.Baltzer, Basel, 1986.

[3] Błażewicz, J., Finke, G., Haupt, R., Schmidt, G., New trends in machine scheduling, European J. of Operational Research 37, No 3, 1988, pp. 303-317.

[4] Carrie, A.S., Petsopoulos A.C., Operations Sequencing in an FMS, Robotica 3, No 4, 1985, pp. 259-264.264.

[5] Chang, Y.L., Sullivan, R.S., Real-time scheduling of FMS, paper presented at the TIMS/ORSA Meeting, San Francisco, CA, 1984.

[6] Coffman, E.G.Jr. (ed.), Computer & Job / Shop Scheduling Theory, J.Wiley, New York, 1976.976.

[7] Erschler, J., Roubellat, F., Thuriot, C., Periodic release strategies for FMS, paper presented at the TIMS/ORSA Meeting, San Francisco, CA, 1984.

[8] Finke, G., Kusiak, A., Modelling and solving the flexible forging module scheduling problem, Eng. Opt. 12, No 1, 1987, pp. 1-12.

[9] Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Optimization and approximation in deterministic sequencing and scheduling theory: a survey, Ann. Discrete Math. 5, 1979, pp. 287-326.

[10] Jaikumar,.R., Postindustrial manufacturing, Harvard Business Review, December 1986.

[11] Jaikumar, R., Van Wassenhove, L.N., A production planning framework for flexible manufacturing systems, Working paper, Harvard University, 1987.

[12] Kusiak, A., Application of operational research models and techniques in flexible manufacturing systems, European J. of Operational Research 24, No 3, 1986, pp. 336-345.

[13] Kusiak, A., (ed.), Flexible Manufacturing Systems: Methods and Studies, North Holland, Amsterdam, 1986.

[14] Kusiak, A., (ed.), Modelling and Design of Flexible Manufacturing Systems, Elsevier, New York, 1986.

[15] Kusiak, A., Scheduling flexible machining and assembly systems, Annals of Operations Research 15, 1988, pp. 337-352.

[16] Kusiak, A., Wilhelm, E., Analysis, Modelling and Design of Modern Production System, Annals of Operations Research 17 J.C.Baltzer, Basel, 1989.

[17] Schmidt, G., CAM: Algorithmen und Decision Support für die Fertigungssteuerung, Springer Verlag, Berlin, 1989.

[18] Srishkandarajah, C., Sethi, S.P., Ladet, P., Scheduling methods for a class of flexible manufacturing systems, Annals of Operations Research 17, 1989, pp. 139-162.

[19] Stecke, K.E., Design, planning, scheduling and control problems of flexible manufacturing systems, Annals of Operations Research 3, 1985, pp. 3-12.

[20] Stecke, K.E., Solberg, J.J., Loading and control policies for a flexible manufacturing system, International J. of Production Research 19, No 5, 1981, pp. 481-490.

[21] Stecke, K.E., Suri, R., (eds.), Flexible Manufacturing Systems: Operations Research Models and Applications, Annals of Operations Research 3, J.C.Baltzer, Basel, 1985.

[22] Stecke, K.E., Suri, R.,(eds.), Flexible Manufacturing Systems: Operations Research Models and Applications II, Annals of Operations Research 15, J.C.Baltzer, Basel, 1988.

[23] Villa, A., Rosetto, S., On the joint problem of dynamic part routing and station service control in flexible manufacturing systems, International J. of Material Flow 2, 1985, pp. 97-110.

[24] Yao, D.D., Material and information flow in flexible manufacturing systems, International J. of Material Flow 2, 1985, pp. 143-149.149

Captions to figures

Fig. 1. An example FMS system

Fig. 2. An example production schedule

Fig. 3. An example vehicle schedule

Fig. 4. Production and non-feasible vehicle schedules

Fig. 5. Optimal production and vehicle schedules

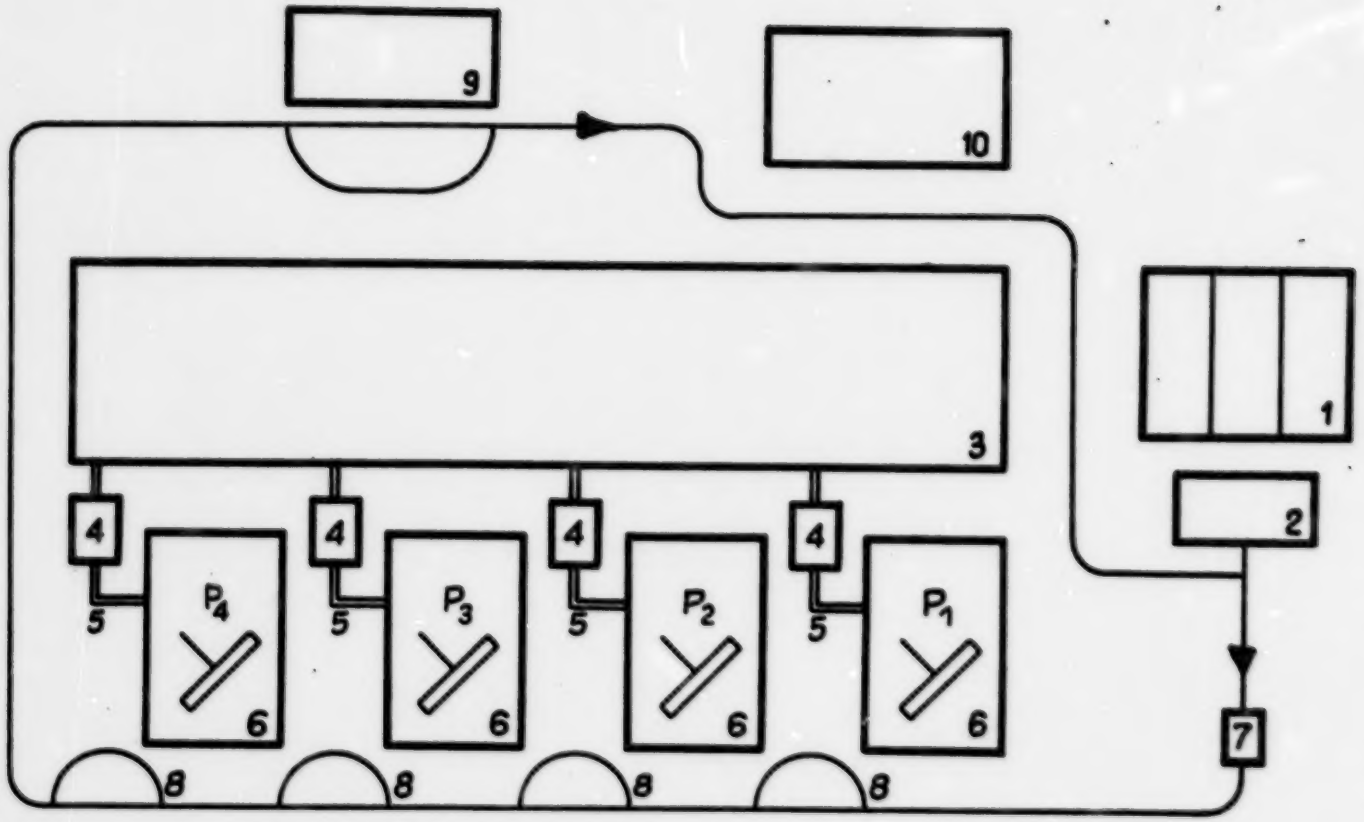


Fig. 1

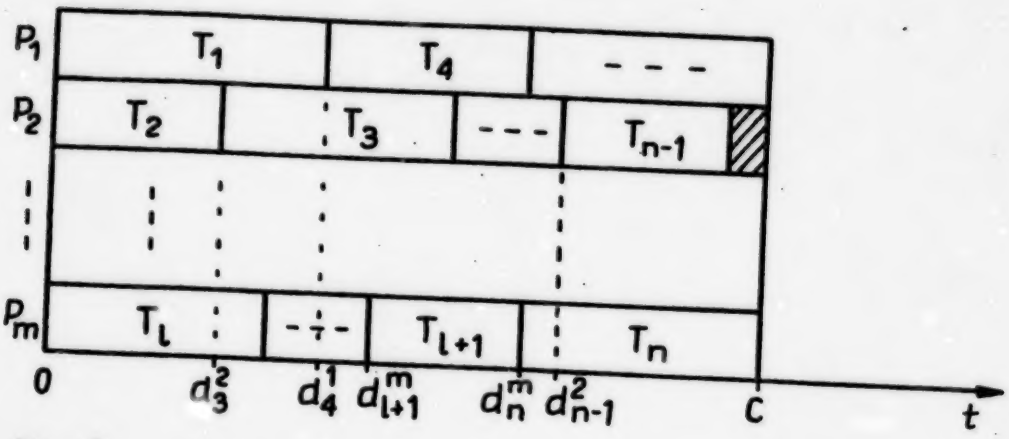

P1
T1
T4
P2
T2
T3
Tn-1
Pm
Tl
Tl+1
Tn
0
C
t

Fig. 2

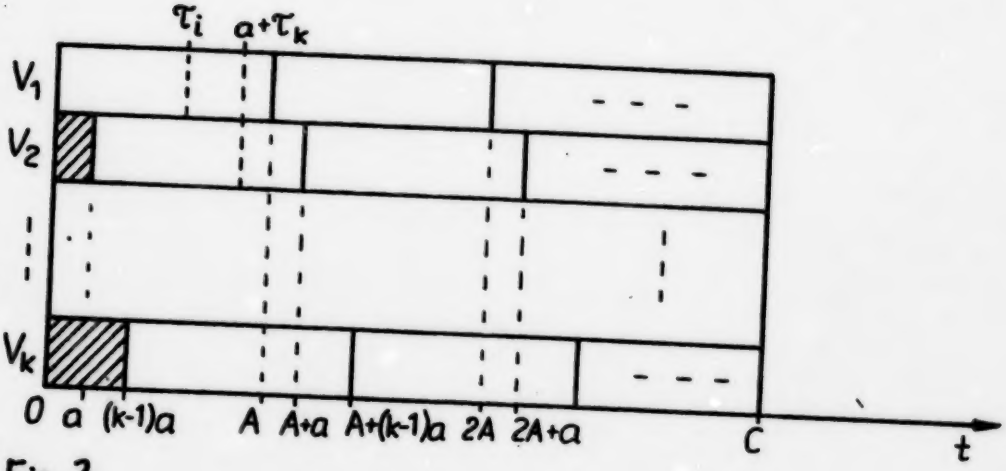
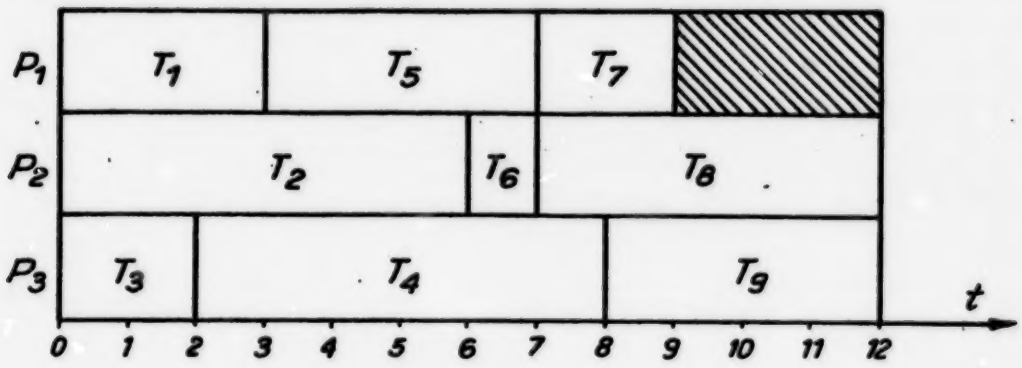
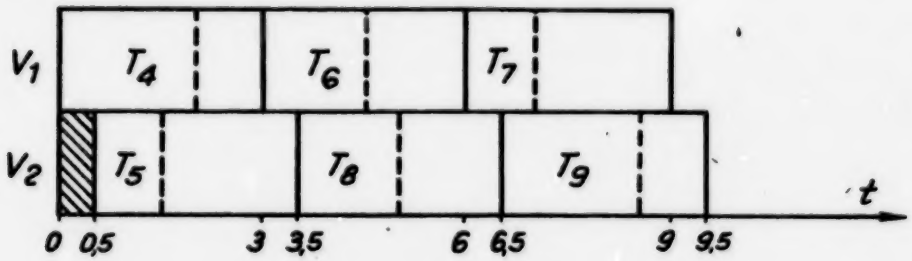

τi
a+τk
V1
V2
Vk
0
a
(k-1)a
A
A+a
A+(k-1)a
2A
2A+a
C
t

Fig. 3

a) Production schedule



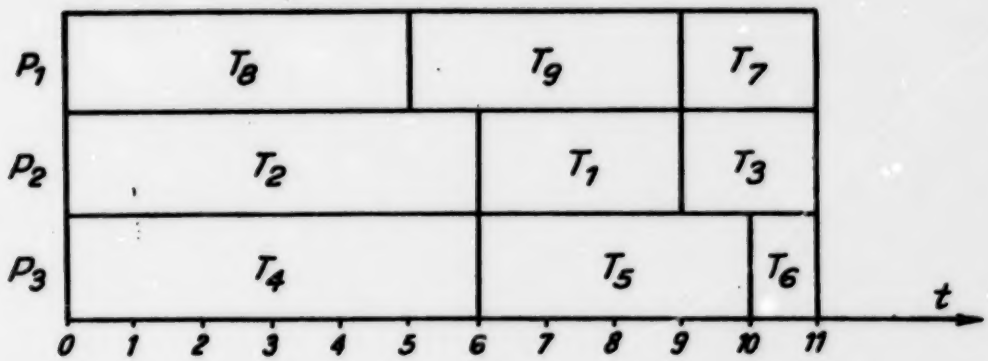
b) Vehicle schedule



T_9 is delivered too late

Fig. 4.

a) Production schedule



b) Vehicle schedule

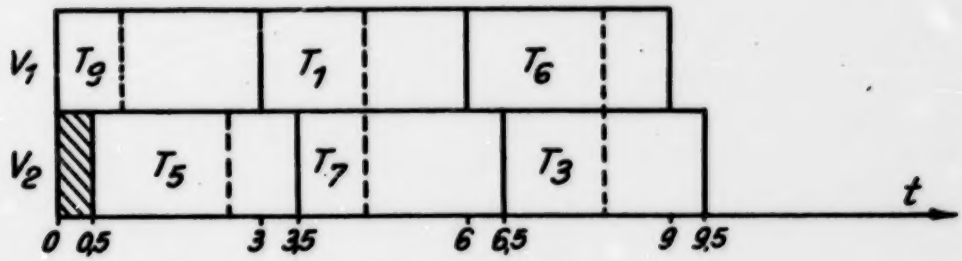


Fig. 5.